Temporal Segmentation of Tasks from Human Hand Motion

Sing Bing Kang and Katsushi Ikeuchi

April 1993

CMU-CS-93-150

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This research was supported in part by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597, and in part by NSF under Contract CDA-9121797.

Keywords: task understanding, human hand motion, motion segmentation, hand tracking.

Abstract

This report describes work on the temporal segmentation of grasping task sequences based on human hand motion. The segmentation process results in the identification of motion breakpoints separating the different constituent phases of the grasping task. A grasping task is composed of three basic phases: pregrasp phase, static grasp phase, and manipulation phase. The pregrasp phase is the initial stage of the grasping task prior to the establishment of a stable grasp (static grasp phase) involved in the task. The manipulation phase refers to the purposeful hand-object interaction performed to achieve a goal in the task. In the pregrasp phase, the trajectory of the hand and the movement of the fingers follow an established pattern. Specifically, it comprises two parallel and simultaneous components: the hand transportation, and the hand preshape.

We show that by analyzing the fingertip polygon area (which is an indication of the hand preshape) and the speed of hand movement (which is an indication of the hand transportation), we can divide a task into meaningful action segments such as approach object (which corresponds to the pregrasp phase), grasp object, manipulate object, place object, and depart (a special case of the pregrasp phase which signals the termination of the task). We introduce a measure called the *volume sweep rate*, which is the product of the fingertip polygon area and the hand speed. The profile of this measure is also used in the determination of the task breakpoints.

The temporal task segmentation process is important as it serves as a preprocessing step to the characterization of the task phases. Once the breakpoints have been identified, steps to recognize the grasp and extract the object motion can then be carried out.

Table of Contents

List of Figures

List of Tables

1 Introduction

Robot programming is an essential component of task automation. The current methods for robot programming include teaching (e.g., [3], [23]), textual programming (e.g., [8], [9]), and automatic programming (e.g., [14], [22], [26], [39]). The first two methods are by far the most pervasive in both the industrial and academic environments. In teaching methods, the robot or manipulator learns its trajectory either through a teach pendant or actual guidance through the sequence of operations ("teach-by-guiding" or less appropriately "teach-by-showing"). This method of teaching is the easiest to use since the implicit knowledge of the task is not necessary. On the other hand, because "teach-by-showing" involves some degree of repetition as a result of errors, it can be tiring and possibly risky. Furthermore, this method is not easily transferable to a different system. Textual programming, while comparatively more flexible, requires expertise and often a long development time.

The problems associated with "teach-by-showing" and textual programming can be alleviated by automatic programming, where conceptually the only inputs required to generate the control command sequences to the robot system are the description of the objects involved in the task, and the high-level task specifications. However, realization of a practical system with automatic programming is difficult in part because of the complexity of path and grasp planning and high-level task goal interpretation (i.e., breakdown of terse high-level goal descriptions into direct actions to be performed to achieve that goal).

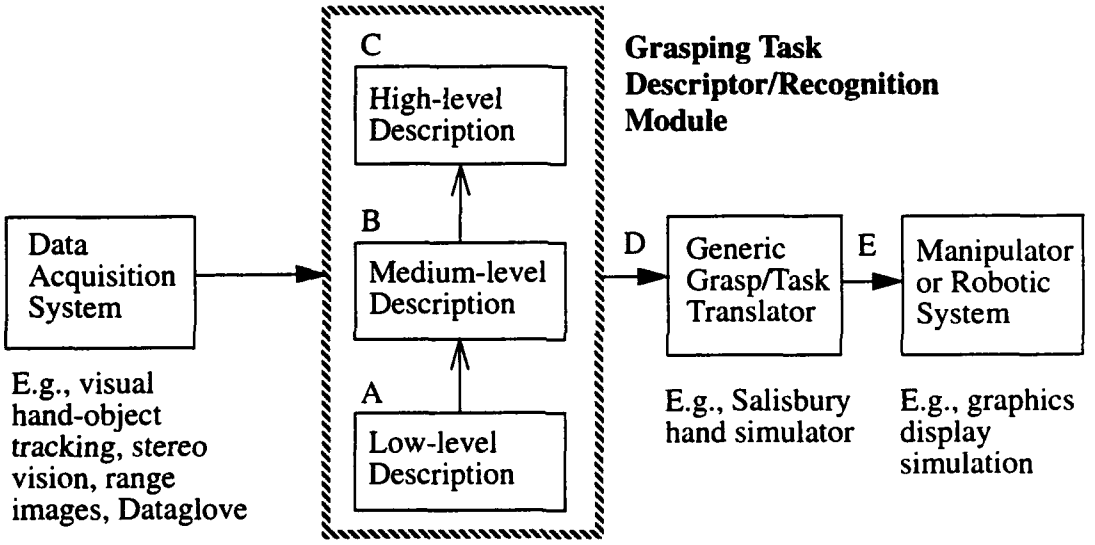
We have earlier stated some of the problems that exist for the more traditional approaches to task programming. We could at least mitigate these problems by using a different approach to task programming. The approach that we adopt in task programming is the *Assembly Plan from Observation* (APO) paradigm proposed by Ikeuchi and Suehiro [15]. In this approach, task programming is performed by demonstrating the task to the system rather than by the traditional method of hand-coding. It allows us to concentrate on understanding hand grasping motions. The key idea is to enable a system to observe a human performing a task, understand it, and perform the task with minimal human intervention. This method of task programming would obviate the need for a programmer to explicitly describe the required task, since the system is able to understand the task based on observation of the task performance by a human.

A similar approach to APO was taken by Kuniyoshi *et al.* [22] who developed a system which emulates the performance of a human operator. However, their system is restricted to pick-and-place operations. Takahashi and Ogata [37] use the virtual reality environment as a robot teaching interface. The operator's movements in the virtual reality space via the VPL dataglove are interpreted as robot task-level operations by using a finite automaton model. Hamada *et al.* [10], on the other hand, specify commands such as "carry(cap, path, body)" to interactively carry out operations. This is first simulated in a "task mental

image" comprising *a priori* action knowledge and a graphical display. Subsequently, the operations are carried out by the manipulator with the aid of a vision system that matches the "mental image" models with the real objects.

1.1 Automatic robot instruction via Assembly Plan from Observation

We adopt the approach of *Assembly Plan from Observation* (APO) in our task programming work. In this approach, the human provides the intelligence in choosing the hand (end-effector) trajectory, the grasping strategy, and the hand-object interaction by directly acting them out. This helps to alleviate the problems of path planning, grasp synthesis, and task specification. Our system, which incorporates the APO paradigm, is snown in Fig. 1.



A, B, C: Hierarchical description modules of grasping task
D: Output description of grasping task
E: Control commands

Fig. 1 **System with perceptual task programming**

The data acquisition system extracts data from the environment; it provides low-level information on the hand location and configuration, objects on the scene, and with some analysis, contact information between the hand and the object of interest. Note that vision need not necessarily be the sole sensing modality through which low-level data is extracted. The grasping task descriptor/recognition module forms the basis of our work. It translates low-level data into higher levels of abstraction to describe both the motion and actions taken in the sequence of operations performed in the task. The vertical arrows in this module as shown in Fig. 1 indicate the consolidation and interpretation of lower-level information to yield successively higher-level information. The low-level description refers to the joint

angles and positions, the medium-level the grouping of functionally equivalent fingers, and the high-level the type of grasp itself [18][19].

In our system, the output of the grasping task descriptor module is subsequently provided to the interpreter (task translator) which in turn creates commands for the robotic system to execute in order to perform the observed task. The representations given in submodules A, B, and C are expected to be independent of the manipulator used in the backend of the system, while the converse is true of the translator.

Our work in this area is expected to result in a greater understanding of grasping motions, to the extent that recognition by a robotic system would be possible. The areas in which this body of knowledge is potentially useful include planning, automation, and teleoperation. Specifically, in our work, since the grasp is described using increasingly abstract and manipulator-independent representations, the resultant system would be conceptually applicable to any given manipulator (which is capable of prehensile grasping) to be used in the robotic system. One potential problem is that the best grasp is dependent on both the shapes and relative sizes of the object and hand. (This is easy to see by comparing the grasps that are used to secure a hold on a medium-sized object and a small object.) We are using the assumption that the relative sizes of the hand and manipulator are comparable, so that an object that can be held comfortably within the compass of the human hand can also be held in a similar manner using the manipulator. If the manipulator is of a disproportionate size relative to the hand, it would not be difficult to scale the size of the object appropriately. (However, the dynamics and control issues would not be as simple.)

We believe that there exists a set of high-level descriptions of tasks relevant to both humans and robots; we are very interested in identifying this common denominator. Among them could well be the high level description of the grasp employed in the grasping task and the description of motion of the hand relative to the object used in the task. This is a major motivating factor for our work on identifying grasps from observation [18] and characterizing the phases of a grasping task.

1.2 Temporal segmentation of a task

This report describes our work on the temporal segmentation of a given task sequence into meaningful parts, namely reaching for the object, grasping the object, and manipulating the object (respectively the pregrasp, grasp and manipulation phases, which are described in greater detail in the next section). The temporal task segmentation is important as it serves as a preprocessing step to identify the frames associated with the phases. This information would then be used to focus on the relevant frames in order to characterize the phases in the task. For example, when the grasp phase has been temporally located, the grasp can then be identified using the location of the object and the hand configuration data [18]. In addition,

by analyzing the motion of the object within the manipulation phase, the type of motion can be extracted and determined.

1.3 Phases in a grasping task

As mentioned in the previous section, there are three identifiable phases in a grasping task:

1.3.1 Pregrasp phase[1]

This is the first phase of the grasping task which precedes the actual grasp. It is a combination of the trajectory of the hand ([2], [16]) (*hand transportation*), and the temporal changes in finger joint parameters in anticipation of the intended grasp ([2], [16]) (*hand preshape*). The trajectory of the hand is influenced by the distance of the object from the hand ([2], [17]) while the finger joint parameter changes are dependent on the shape of the object ([2], [17], [39]). The hand transportation and hand pre-shape components have been observed to occur in parallel. Features of the hand pre-shape such as the approach area, approach volume, and approach axis, as described in [27], can be used to characterize this stage.

1.3.2 Grasp phase

The pregrasp phase ends and the static grasp phase begins at the moment the hand touches and has a stable hold of the object. The type of grasp employed can be identified at this phase, and can be represented using a grasp hierarchy proposed by Kang and Ikeuchi [18].

1.3.3 Manipulation phase

The manipulation phase is characterized by hand motions resulting in the purposeful movement of the object relative to the environment. The grasp is chosen by the operator on the basis of the mobility and dexterity required to manipulate the object.

A manipulative action can be as simple as just translating the object with respect to the environment. It can be as complex as simultaneously transporting (by hand transportation) and precision handling ([24], [25]) the object with the fingertips, changing its pose with respect to both the palm and the environment.

The idea of a *homogeneous* manipulation to describe the smooth object motion while perturbing a single static grasp is introduced in [33]. A complete task may comprise several homogeneous manipulations. Perlin et al. [33] propose a structured and hierarchical

1. This has been variously referred to as reaching ([2], [5], [39]) and target approach ([35], [39]). However, these terms can be easily confused with the hand transportation component of this phase - Jeannerod [16], for example, uses the terms reaching and transportation interchangeably.

approach to autonomous manipulation, specifically for the Utah/MIT hand. The scheme involves the establishment of the static grasp taxonomy, from which a library of homogeneous manipulations and subsequently low-level control primitives and sensor interactions may be developed. In our work, we assume homogeneous manipulation (i.e., the same grasp is employed) within a manipulation phase.

Hirai and Sato [12] developed a system capable of recognizing a slave robot's pick-and-place motions based on its joint sensors and force sensors attached to its two fingers. The end-effector is a parallel-jaw gripper with a 6 DOF force sensor at the base of each finger. The robot motion understanding is based on rules with pre- and post-conditions; a motion state is recognized if the pre-conditions are satisfied. The primary motions that the system can recognize are the approach, move-to-grip, and grip motions.

Pook and Ballard [34] use finger tendon tensions of a teleoperated Utah/MIT hand to recognize specific teleoperated actions such as grasping a spatula, carrying the spatula, press the spatula against the pan bottom, and sliding the spatula along the pan surface. The basic method of recognition is pattern classification via vector quantization, k-nearest neighbor classification and Hidden Markov Model.

Hashimoto and Buss [11] describe a system with a stationary sensor "glove" which measures the finger joint angles and exerts forces on the hand according to its simulated interaction with the virtual object. They model the manipulative skill using a time-based sequence of the grip transformation matrix proposed by Salisbury [31].

1.4 Organization of report

Section 2 reviews some of the characteristics of the pregrasp phase as described in the literature of human hand movement. The important features that are used in characterizing hand motion are discussed at greater length. In Section 3, we describe the proposed task segmentation algorithm using these features. Results of the task segmentation algorithm on several tasks are also shown and discussed. We show in Section 4 how the results of the segmentation algorithm can be used to further identify the type of grasp employed in the task and extract object motion during the manipulation phase. Finally, a summary of our work is given in Section 5.

2 Features for Segmentation of a Task Sequence

In this section, we review research on human hand movement, specifically on the characteristics of reaching motions of the hand (i.e., during the pregrasp phase). Subsequently, we describe the features that are used in our framework of task segmentation. Relevant work on motion representation are also briefly discussed.

2.1 Studies in human hand movement

Numerous studies on human hand movement point to commonly established characterizations of the pregrasp phase. The pregrasp phase has been analyzed in terms of two simultaneous activities, namely the hand reaching activity (termed the *hand transportation* component), and the finger activity in anticipation of the grasp (termed the *hand preshape* component[1]) (e.g., [17], [28], [29], [41]).

Typical profiles of the hand transportation speed and grip aperture[2] during the pregrasp phase are shown in Fig. 2. The characteristic inverted bell-shaped curve of the hand transportation speed has been observed by many researchers (e.g., [16], [32]).

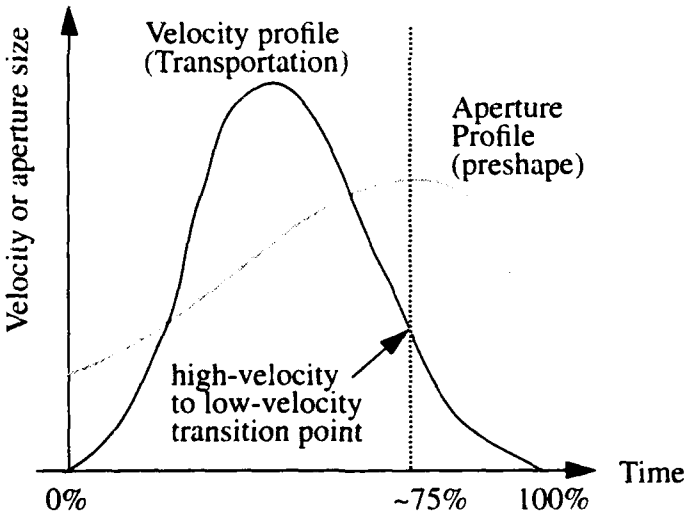


Fig. 2 **Typical pregrasp component profiles**

Jeannerod ([16], [17]) conducted experiments involving reaching and grasping movements to see how object characteristics affect these movements. His experiments show that object size and orientation affect hand preshape but not hand transportation. However, object dis-

1. This has also been referred to as the grasping or manipulation component. The alternative terms are not used here to avoid confusion with the static grasp phase and the manipulation phase of the grasping task.

2. The grip aperture in this context is defined as the separation between the tips of the thumb and index finger.

tance influences only the hand transportation and not the hand preshape. Another interesting inference from his series of experiments is the temporal coincidence between the starting of the hand preshape aperture reduction and the commencement of the low-velocity phase of the hand tranportation. (The point at which the low-velocity phase begins is at the time where the lowest acceleration occurs.) These happen almost simultaneously after about 75% of movement time had elapsed. Fig. 3 shows the temporal divisions of the hand preshape and hand transportation components into different subphases whose boundaries coincides.

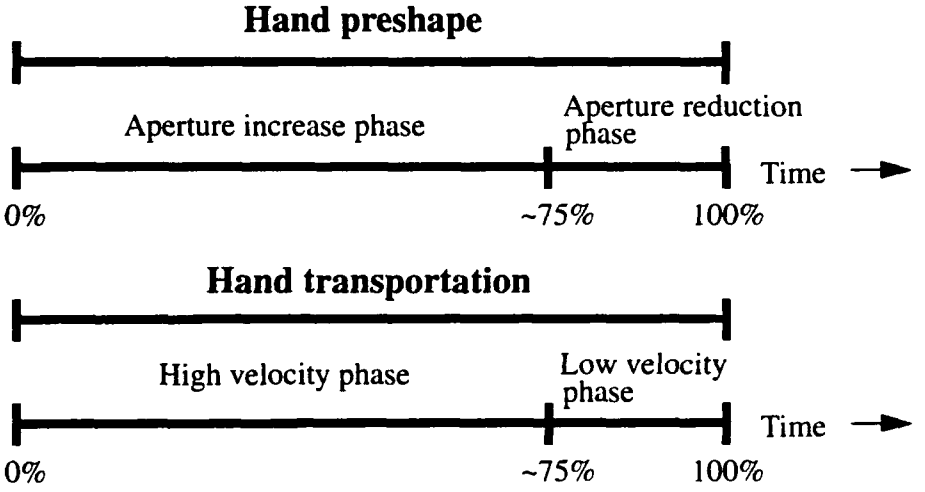


Fig. 3 **Pregrasp phase components**

The hand preshape component is controlled by the distal muscles of the body and appears to be activated by intrinsic or object-centered properties such as object shape and size [16]. In contrast, the hand transportation component is controlled by proximal muscles and appears to be activated by extrinsic or viewer-centered properties such as object location relative to the person [16].

Marteniuk and Athenes [29] found that the maximum grip aperture has very strong linear correlation with the object size and that movement time increased linearly with the decrease in object (disk) size. The latter result is due to the increase in the duration of hand deceleration while the duration of hand acceleration remained constant. The ratio of these two times could perhaps be linked to the precision requirements of hand motion in the task, as Marteniuk et al. [30] suggest. Marteniuk and colleagues [30] noted that the objective of a task (which dictates the precision requirements of the task) affect trajectory shape. For example, in one of their experiments, the duration of hand deceleration was disproportionately longer for the task of fitting a disk into a well when compared to that for a task of picking up a disk and throwing it into a large box.

Wing, Turton and Fraser [41] report that the grasp aperture (separation between tips of the thumb and index finger) was greater in cases where reaching movements were performed

faster and where there was no visual feedback (i.e., subjects had their eyes closed while reaching for the object).

The results of the research on human hand motion point to the importance of both the grip aperture and speed of the hand in characterizing the pregrasp phase. These studies that highlight the characteristic shapes of the grip aperture and speed profiles indicate that these measures may be used to temporally segment a task sequence into its constituent phases. Both of these metrics (in one form or the other) form the bases of our work on temporal task segmentation. Three quantifiable measures that are proposed are the *hand volume*, the *fingertip polygon*, and the *fingertip polygon normal.*

2.2 The hand volume, fingertip polygon, and fingertip polygon normal

Lyons [27], in describing a conceptual high-level control mechanism for a dexterous hand, defines the following terms:

- approach volume - the volume between the fingers
- approach area - surface formed by joining the fingertips of the preshaped hand by straight lines
- approach axis - outward normal to the approach area through its centroid

Lyons uses these terms in the context of the pregrasp phase. We extend these definitions to the manipulation phase as well; the corresponding terms that we use are:

- hand volume
- fingertip polygon area
- fingertip polygon normal

Fig. 4 depicts the ideas of hand volume, fingertip polygon, and fingertip polygon normal. These features are potentially useful in characterizing a task. In fact, the fingertip polygon is used as one of the primary features for temporal segmentation in our framework.

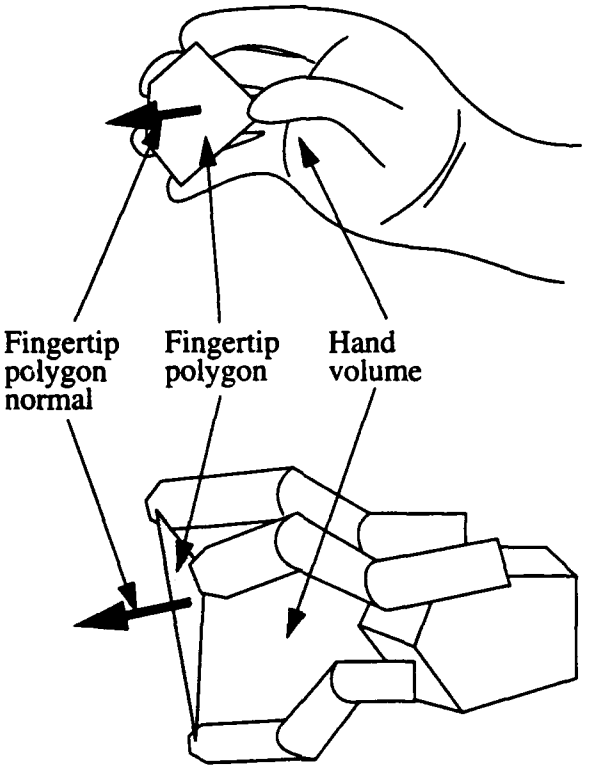


Fig. 4 **Fingertip polygon normal, fingertip polygon, and hand volume for the hand (top) and a *manipulator (bottom)***

2.2.1 Calculating the area and centroid of the fingertip polygon

Wing and Fraser [40] found from their experiments that the thumb contributed significantly less in the reduction of grasp aperture that the other fingers. They suggest that the relative stability of the thumb is due to its role in guiding the hand transportation of the pregrasp phase. In light of this research, it would seem reasonable that the position of the centroid within the fingertip polygon would be more heavily influenced by the position of the tip of the thumb.

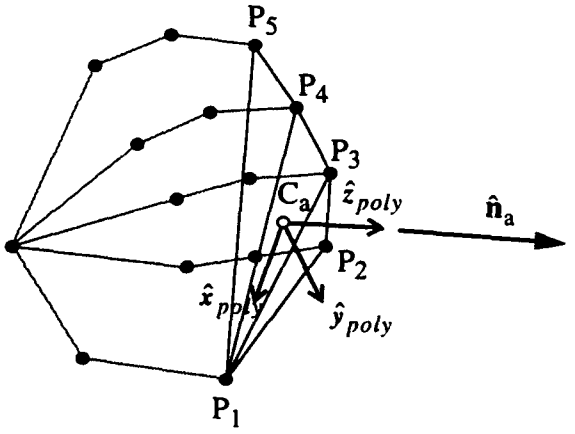


***Fig. 5* Contact web [18], the fingertip polygon and the fingertip polygon normal and coordinate frame**

Consider the contact web representation [18] of the hand at a given point in time during the pregrasp phase (shown in Fig. 5). The fingertips are denoted as P_1 (tip of the thumb), P_2, P_3, P_4 and P_5; C_a is the centroid of the fingertip polygon while $\hat{\mathbf{n}}_a$ is the fingertip polygon normal. The area of the fingertip polygon is

$$A_{app} = \sum_{k=2}^{4} A_k \tag{1}$$

where A_k is the area of the triangle $P_1P_kP_{k+1}$ (calculated using Heron's formula):

$$A_k = \sqrt{s_k (s_k - l_k)\ (s_k - l_{k+1})\ (s_k - m_k)} \tag{2}$$

where

$$s_k = \frac{1}{2}(l_k + l_{k+1} + m_k) \tag{3}$$

l_k is the distance between P_1 and P_k, and m_k is the distance between P_k and P_{k+1}.

The centroid of the fingertip polygon is

$$C_a = \frac{1}{4}\sum_{i=2}^{5} \frac{P_1 + P_i}{2} = \frac{\sum_{i=1}^{5} \mu_i \omega_i P_i}{\sum_{i=1}^{5} \mu_i \omega_i} \tag{4}$$

where $\omega_1 = 4$ and $\omega_i = 1$ for $i = 2, ..., 5$. $\mu_i = 1$ if finger i is involved in the static grasp phase and 0 otherwise.

The fingertip polygon normal is taken to be the normal of the best fit plane to the fingertips (away from the hand). The frame origin of the fingertip polygon is at its centroid with the x-axis defined to be the unit vector pointing towards the thumb fingertip and the z-axis defined as the normal (Fig. 5).

A very important question arises: when do we know that the object has been grasped and at which point does the object move with the hand? One possibility is to use explicit motion boundaries.

2.3 Motion representation using explicit boundaries

Rubin and Richards [36] propose to characterize visual motion using explicit boundaries that they define as starts, stops and force discontinuities (step and impulse). When one of these boundaries occurs in a motion, human observers have the subjective impression that some fleeting, significant event has occurred. Iba [13] augments these elementary boundaries with zero crossings in accelerations. While these motion boundaries show promise in task segmentation, they are prone to noise and are less reliable when the sampling rate is low, as was the case in our experimental setup.

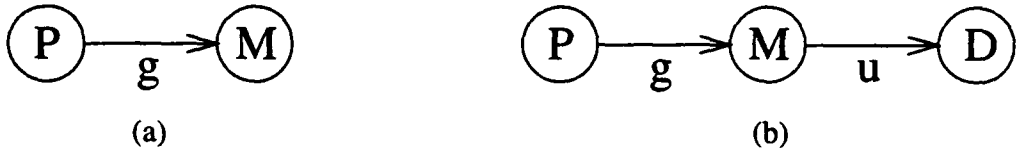
3 Temporal Segmentation of a Task Sequence

In this section, we first define the notions of a *subtask* and an *N-task*, and describe how a task can conveniently be pictorially represented as a state transition diagram. We then describe the task segmentation algorithm. Subsequently, we show, with examples, how the identified task breakpoints that separate the different phases can be used to identify the grasp and extract the object motions in the manipulation phase. Determining the object motions in the manipulation phase is useful in identifying the actions performed on the object during the task.

3.1 State transition representation of tasks and subtasks

An assembly task may comprise a variety of operations such as moving towards an object, picking up an object, moving the grasped object, inserting one object onto another, etc. It is convenient to represent the task as a series of states and transitions. As mentioned earlier, a task unit (called a *subtask* from this point on) is composed of three phases, namely the pre-grasp, grasp, and manipulation phases. The grasp phase is treated as a transition from the pregrasp phase to the manipulation phase. The state transition diagram for a subtask is shown in Fig. 6(a). A task would minimally comprise these three phases and an ungrasp and depart motions. The ungrasp motion is the opposite of the grasp motion while the depart

phase is a specialized instance of the pregrasp phase. Hence a task has minimally one embedded subtask; such a task is referred to as a *1-task* (Fig. 6(b)). A task with *N* subtasks is termed an *N-task*. The state transition diagram for a general task is depicted in Fig. 7.



(a) (b)

States:
P: pregrasp phase
M: manipulation phase
D: depart phase

Transitions:
g: grasp
u: ungrasp

***Fig. 6* State transition diagram of a (a) subtask and (b) 1-task**

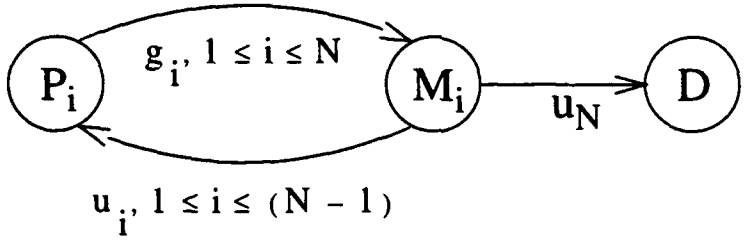


***Fig. 7* State transition diagram of a general task (N-task)**

The state transition diagram representation lays the groundwork for the temporal division of a task sequence and facilitates the visualization of the extracted task components.

3.2 Temporal segmentation of task into subtasks

We can segment the entire task into meaningful subparts (such as the different states and transitions described in the previous section) by analyzing both the fingertip polygon area and the speed of the hand. The fingertip polygon area is an indication of the hand preshape while the speed of the hand is an indication of the the hand transportation. While a viable alternative appears to the grip aperture, i.e., the distance between the thumb and the index finger, this feature is more prone to sensor error and uncertainty.

Intuitively, in the pregrasp phase, as a person moves his hand towards an object with the goal of picking it up, he unconsciously increases the spacing between the fingers in anticipation of the grasp. This yields the characteristic inverted bell curve profile of the fingertip polygon area during this time. The speed profile of the hand also assumes this trend, due to the initial acceleration and the subsequent deceleration. Once the object has been picked and

is being moved, we arrive at the manipulation phase of the task. Assuming homogeneous manipulation, the fingertip polygon area remains approximately constant. Once again the speed profile of the hand assumes the inverted bell curve of acceleration and deceleration. By taking into consideration both the fingertip polygon and speed profiles in the pregrasp and manipulation phases, we can more reliably divide the entire task into the following actions: reach for object, grasp object, move or manipulate object, and place object. The breakpoints can be extracted more reliably in this manner than from just the speed or fingertip polygon profile alone. This can be seen by considering the speed and fingertip polygon profiles in Fig. 13 which possess many local minima. The significant values of the fingertip area during the first manipulation phase (frames 11-17) also complicates the segmentation process.

A useful profile to analyze is the profile of the product of the speed and fingertip polygon area at each frame called the *volume sweep rate* profile. The physical interpretation of the volume sweep rate is illustrated in Fig. 8. It measures the rate of change in both the fingertip polygon area and speed in 3D space. While the hand reaches to grasp the object, both the speed and fingertip polygon area profiles are bell-shaped, and although the peaks are not coincident, its volume sweep rate has an accentuated peak, and hence a comparatively higher peak value. This can be seen from the graph based on typical real data in Fig. 9. Meanwhile, during object manipulation, the fingertip polygon area is always less than the peak fingertip polygon area of the reach-object phase just prior to the grasp. This results in a smaller peak value of its volume sweep rate. The volume sweep rate profile is used to get rid of local extrema points (minima) in the speed profile that are not task breakpoints.

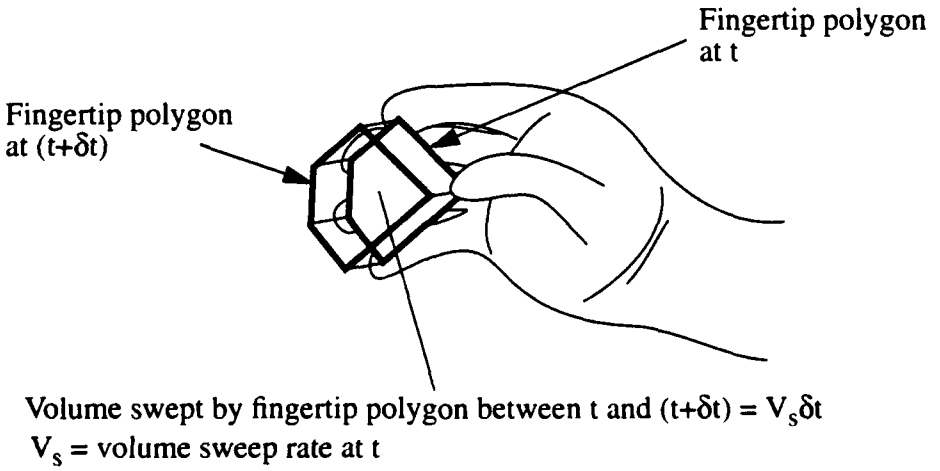


Fig. 8 **Physical interpretation of volume sweep rate**

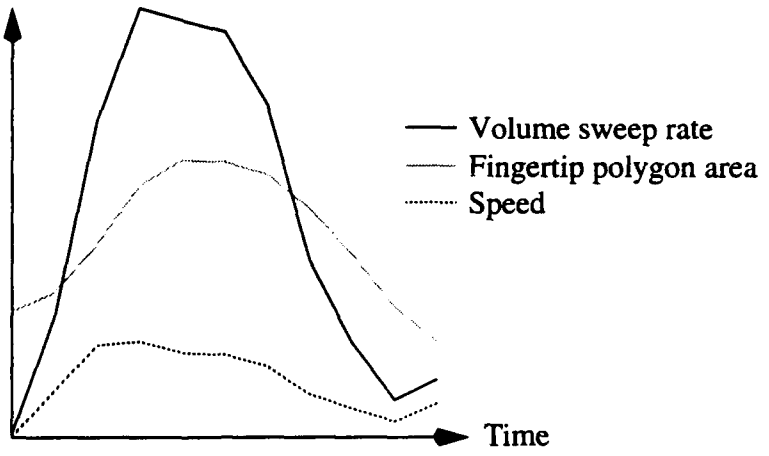


Fig. 9 **Typical profiles of fingertip polygon area, hand speed, and volume sweep rate during the pregrasp phase**

The algorithm to segment a task sequence into meaningful subsections starts with a list of breakpoints comprising local minima in the speed profile. The initial breakpoints are extracted from the speed profile rather than the volume sweep rate profile as while the latter is useful in globally locating the true breakpoints, it contains more local minima. The global segmentation procedure makes use of:

1. *The condition that the pregrasp phases and the manipulation phases interleave;*
2. *The condition that the peak of the volume sweep rate in the manipulation phase is smaller than those of the two adjacent pregrasp phases;*
3. *The condition that the mean of the volume sweep rate in the pregrasp phase is larger than those of the two adjacent manipulation phases; and*
4. *The goodness of fit of the volume sweep rate profiles in the pregrasp phases to parabolas. In fitting the curve to the parabola, the search is pruned if either the estimated peak is out of range of the interval of interest, or if the estimated parabola does not assume an inverted U-shape (the latter is done by checking the estimated equation parameters).*

Let

I_i = interval between breakpoints i and i+1;

N_I = number of hypothesized intervals;

N_M = number of hypothesized manipulation phases = $\left\lfloor \frac{N_I + 1}{2} \right\rfloor - 1$;

$M_{VSR,i}$ = mean volume sweep rate in I_i;

$M_{APA,i}$ = mean fingertip polygon area in I_i;

$F_{VSR,i}$ = root of sum of squared error in parabola fitting the volume sweep rate profile in I_i;

$F_{APA,i}$ = root of sum of squared error in parabola fitting the fingertip polygon area profile in I_i; and

$$D_i = \begin{cases} F_{VSR,1} F_{VSR,1} \dfrac{M_{APA,1}}{M_{VSR,1}} & , i = 0 \\ \dfrac{1}{2}\left(F_{VSR,2i-1} F_{APA,2i-1} + F_{VSR,2i+1} F_{APA,2i+1}\right) & , 1 \le i \le N_M \\ F_{VSR,N_I} F_{VSR,N_I} \dfrac{M_{APA,N_I}}{M_{VSR,N_I}} & , i = N_M + 1 \end{cases}$$

D_i essentially yields the weighted sum of the RMS errors of parabolic fitting of the pregrasp profiles adjacent to the hypothesized manipulation phase. The weight is taken to be the mean polygon area in the pregrasp phase. The objective function associated with the list of breakpoints is given by the mean

$$E = \frac{1}{N_M + 2} \sum_{i=0}^{N_M+1} D_i$$

The desired breakpoints are obtained by minimizing E. Using the volume sweep rate profile rather than the speed profile reduces the tendency to incorrectly group adjacent phases, since the more pronounced peaks in the pregrasp phase make them much more difficult to erroneously classified together with the adjacent manipulation phases.

Given a tentative list of breakpoints, the algorithm tries all the combinations subject to items 1 to 4 above. Many possibilities are pruned by the conditions in items 2, 3 and 4 above; the combination which passes this test and yields the best overall parabola fit is then deemed to be the desired task breakpoints.

3.3 Experiments

3.3.1 Implementation of hand tracking system

We track the configuration (joint angles) and pose (position and orientation) of the hand using the *Cyber*Glove [7] and Polhemus [1] devices respectively. The *Cyber*Glove is an instrumented, lightweight, flexible glove produced by Virtual Technologies. It has 18 sensors (3 flexion sensors for the thumb and 2 for the other fingers, 4 abduction sensors, 1 pinky rotation sensor, and the wrist pitch and yaw sensors). The distal interphalangeal joint angles

are not measured but estimated instead based on the theoretically derived and empirically t :ted relationship between the distal and proximal interphalangeal joint angles [6]. The Polhemus 3Space Isotrak sensing device is attached to the dorsal side of the wrist, and provides the position as well as the orientation of the hand relative to the Isotrak source. The Ogis light-stripe rangefinder and a CCD camera provide the range and intensity images, respectively.

The software which reads in and interprets the hand configuration and pose are mostly written in C (some of which are adapted from the VirtualHand v1.0 software supplied by Virtual Technologies), while that which provides the object representation and its relationship with the hand is written in Common Lisp. The geometric modeler used in our work is Vantage [4]. In addition, the frames created to represent the grasp hierarchy [19] are done using Knowledge Craft [21]. Knowledge Craft is a toolkit for knowledge engineers and AI system builders, and it uses a frame-based knowledge representation language called CRL (Carnegie Representation Languange) with procedural attachment and inheritance.

3.3.2 Experimental results

The temporal task segmentation algorithm has been successfully applied to several task sequences, and the results of two of the sequences can be seen in Fig. 12 and Fig. 13. The breakdown and classification of the frames are shown in Fig. 10 and Table 1 (task sequence 1), and Fig. 11 and Table 2 (task sequence 2). Task sequence 1 (a 3-task) involves only pick and place actions while task sequence 2 (a 4-task) involves pick and place, insert, and screwing actions. For both sequences, the algorithm has correctly identified the frames where an object is grasped and placed, regardless of whether the object is picked and placed, inserted into another object, or screwed into another object.

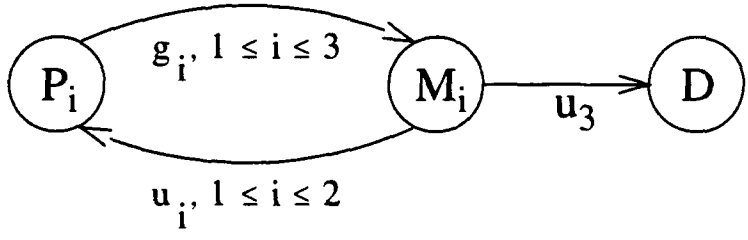


Fig. 10 **State transition representation of task sequence 1 (3-task)**

Table 1 Classification of frames in task sequence 1

i	P_i	g_i	M_i	u_i	D
1	{0, ..., 8}	{9}	{10, ..., 13}	{14}	

Table 1 Classification of frames in task sequence 1

i	P_i	g_i	M_i	u_i	D
2	{15, ..., 18}	{19}	{20, ..., 25}	{26}	
3	{27, ..., 30}	{31}	{32, ..., 37}	{38}	
					{39, ..., 44}

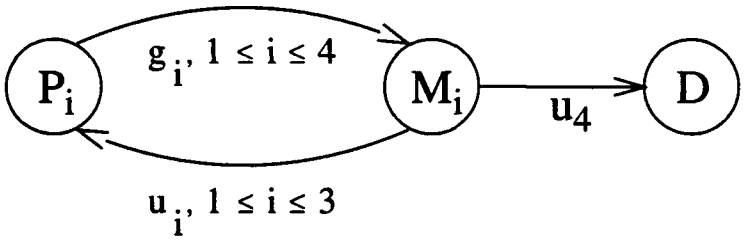


Fig. 11 **State transition representation of task sequence 2 (4-task)**

Table 2 Classification of frames in task sequence 2

i	P_i	g_i	M_i	u_i	D
1	{0, ..., 9}	{10}	{11, ..., 17}	{18}	
2	{19, ..., 23}	{24}	{25, ..., 35}	{36}	
3	{37, ..., 43}	{44}	{45, ..., 51}	{52}	
4	{53, ..., 58}	{59}	{60, ..., 90}	{91}	
5					{92, ..., 101}

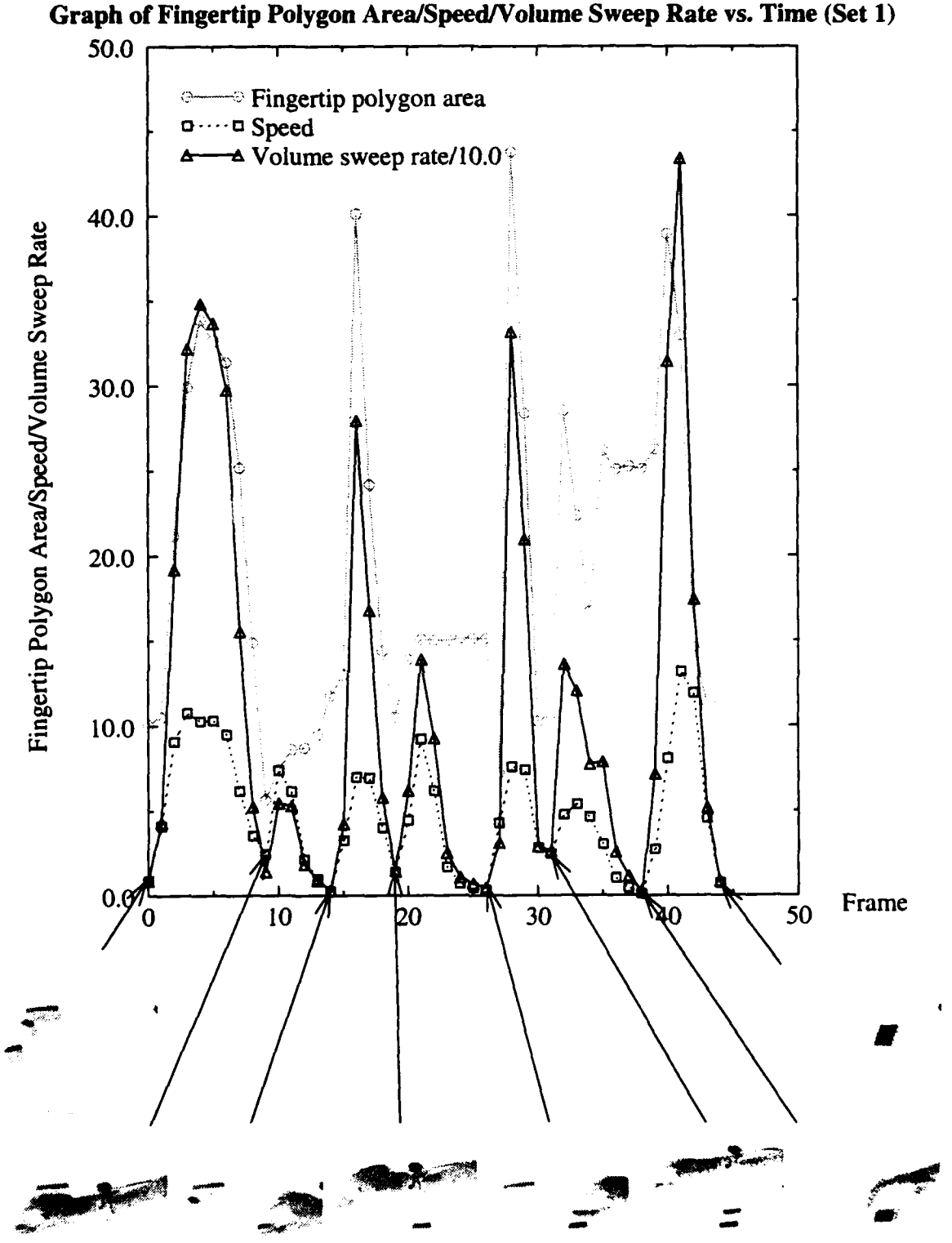


Fig. 12 **Identified breakpoints in task sequence 1 (a 3-task).**

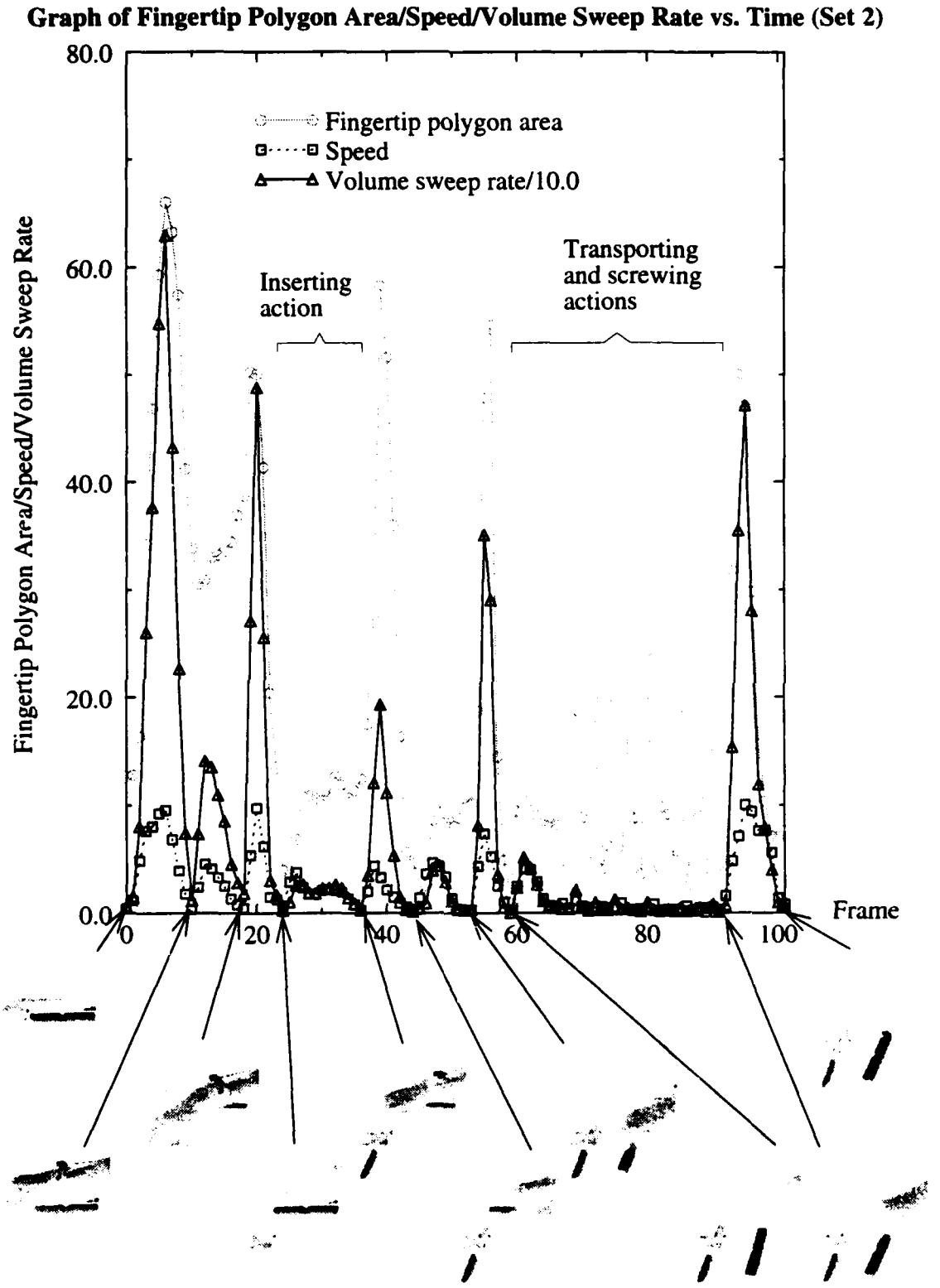


Fig. 13 **Identified breakpoints in task sequence 2 (a 4-task).**

3.4 Implementational problems

After the task breakpoints have been identified, we can then proceed to identify the grasp and extract the object motions during the manipulation phase. However, these two processes are complicated by, among others, errors in the Polhemus readings which cause the raw data of the hand position and configuration to be unreliable. The sources of the errors are:

1. *Distortion of magnetic field by nearby ferromagnetic material. This effect is significant because the Polhemus device uses ac magnetic field technology to determine the relative pose of the sensor to the source.*
2. *Inaccuracies in localization of the Polhemus device in the range image during rangefinder-to-Polhemus frame transform calibration. The inaccuracies are in part attributed to the model built in the geometric modeler not exactly corresponding to the actual shape and specified stepsize and resolution of the fitting algorithm.*
3. *Inaccuracies in the modeling of the hand.*
4. *Misalignment of the Polhemus sensor relative to the hand. This happens as the sensor is not rigidly fixed to the glove.*
5. *Misalignment of the Polhemus source relative to the table, since it is not very rigidly affixed to it.*

Since a task can be broken down into its constituent subtasks, we can then analyze each subtask individually. The subtask analysis involves grasp identification and object motion extraction; we illustrate this analysis with experiments involving 1-tasks.

4 1-task Analysis

Analyzing subtasks is equivalent to analyzing 1-tasks, since each 1-task contains a subtask; there is no loss in generality in illustrating the analysis using 1-tasks. Each subtask can be characterized in terms of the grasp used and object motion during the manipulation phase. The grasp can be identified from contact information between the hand and object at the grasp frame identified by the task segmentation algorithm. This is done by mapping the low-level contact information such as the contact position and normal into increasingly abstract entities such as functionally equivalent groups of fingers and the degree of interaction between these groups [18]. The method to determine the object motion is described in subsection 4.2.

A series of experiments featuring 1-tasks were conducted as follows:

1. *Take the range image of the scene before the subtask. This is used to determine the location of the object of interest.*

2. *Perform the 1-task (which comprises only a set of pregrasp, grasp, and manipulation phases) while its intensity image sequence and the CyberGlove and Polhemus readings are being recorded.*
3. *Take the range image of the scene after the 1-task has been performed.*

This approach was used because with it is not possible to sample range images rapidly with the current setup. The light-stripe rangefinder takes about 10 seconds to cast 8 stripe patterns and calculate the range values for a 256x240 image. The intensity images (each or resolution 128x120) taken during the performance of the 1-task are sampled at a rate of about 1.5 Hz.

4.1 Task segmentation, grasp identification and manipulative motion extraction for 1-tasks

A major problem in identifying the grasp is the imperfect positional information obtained from a real data acquisition system such as ours. In addition, the exact moment of grasping cannot be pinpointed due to the discrete sampling of the hand location and configuration. As a result, we have to resort to extra preprocessing to accommodate such data imperfections, specifically adjusting the orientation of the hand at the grasp frame.

The processes of segmenting the task and determining the grasp and manipulative (i.e., object) motions are done using a three-pass approach. The first pass establishes the motion breakpoints while the second pass involves adjusting the pose of the hand and subsequently determining the grasp employed in the 1-task. Finally, the effect of the reorientation of the hand is propagated throughout the 1-task sequence and the object motion is then extracted using the approach delineated in the following subsection. The details of the three-pass approach are as follow:

Pass 1:

1. *Estimate pose of object from the before-task range image.*

   The initial gross position (but not the orientation) of the object of interest is determined by subtracting the 3D elevation map of the scene after the task from that before the task. The 3DTM[1] program is then used to localize the object. Two refinements were made: (a) use three orthogonal initial poses and pick the final estimated pose with the least RMS fit error; and (b) use coarse-to-fine stepsizes.

2. *Calculate the motion profiles (speed, fingertip polygon area, and volume sweep rate).*
3. *Determine the motion breakpoints from the motion profiles as described earlier.*

---

1. Short for 3D template matching. See Appendix for a brief description of this 3D object localization program.

Pass 2:

1. *From known motion breakpoints (determined in Pass 1), calculate the object motion associated with the manipulation phase (which is bordered by the grasp and ungrasp transitions).*

2. *At the grasp frame, determine the grasp employed.*

   Due to the errors in the Polhemus and CyberGlove readings, the oriented hand may intersect the object. The hand is reoriented (subject to the fixed position of the Polhemus sensor) until: no interpenetration between the hand and object occurs; and the weighted sum of distances between the hand contact points and the object is minimized.

   The determination of the "optimal" hand pose is done with direct search with rotational increments of 1.15^{o} and limited to a maximum of 60^{o} rotation about discretely sampled axes (80 directions sampled on a once-tesselated icosahedron). (An increment of 1.15^{o} would produce at most an error of about 2 mm at a point 10 cm away from the rotation center.)

   The object is stored as a collection of oriented surface points (position and normal information associated with each point) whose spacing is typically between 4.0-7.5 mm. This spacing of the object depends on the object size - it is increased for a larger object size. The nearest distance of each hand contact point to the object is then estimated using this oriented point representation.

   Once the "optimal" pose of the hand and the object-contact information have been extracted, the grasp is then recognized using the classification scheme described in [18].

Pass 3:

1. *Propagate adjustment in both distal and proximal motions throughout the task due to hand reorientation in Pass 2.*

2. *The gross after-the-task pose of the object is determined by successively applying the distal transformations in the frames composing the manipulation phase to the original object position (i.e., prior to the task). This after-the-task pose is refined using the 3DTM program.*

4.2 Determining object motion during the manipulation phase

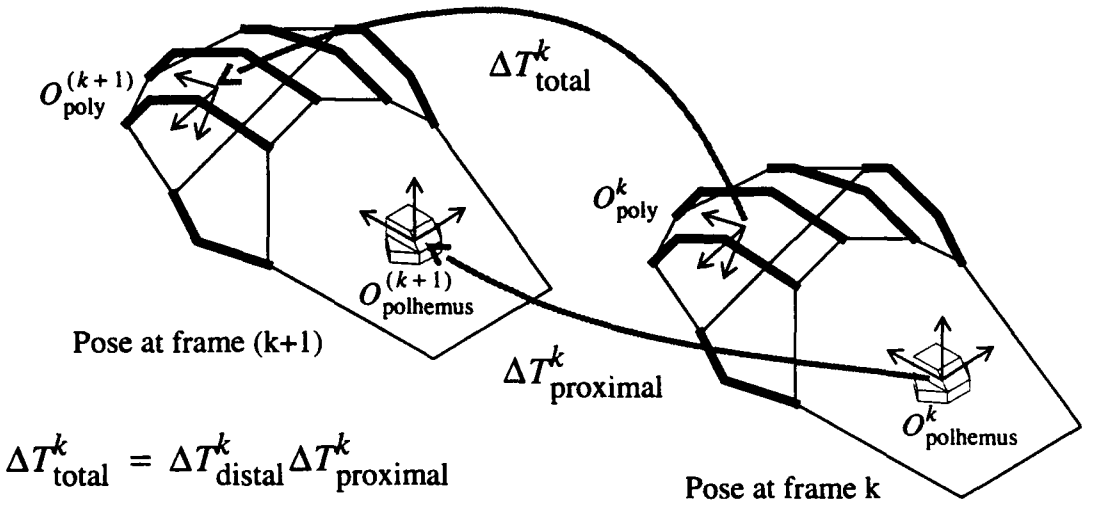


Fig. 14 **Total and proximal motions from frame k to k+1 during the manipulation phase**

It may be useful to determine the *proximal motion* (which corresponds to the motion of the arm and wrist) and *distal motion* (which corresponds to motions of the fingers, otherwise referred to as "precision handling" [24]). The *total motion*, which is the overall effect of both the proximal and distal motions, directly yields the object motion. Meanwhile, the proximal and distal motions yield information on which component of the hand/arm motion is contributing to the object motion.

We can determine the object motion transformations (i.e., the total motion) in the manipulation phase once we have identified the task motion breakpoints. Suppose the kth frame has been identified as the grasp frame and the lth frame the ungrasp frame in the task sequence of N frames. The desired object change in pose at frames between k and l (i.e., during the manipulation phase) can be be determined (Fig. 15) from (5):

$$\Delta T_{k,\,k+j} = T^{k+j}_{hand} (T^{k}_{hand})^{-1} \tag{5}$$

where T^{k}_{hand} is the total transform associated with the motion of both the fingers and hand at the kth frame.

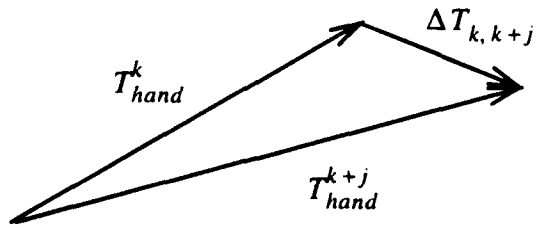


Fig. 15 **Determining the differential motion between two frames in the manipulation phase**

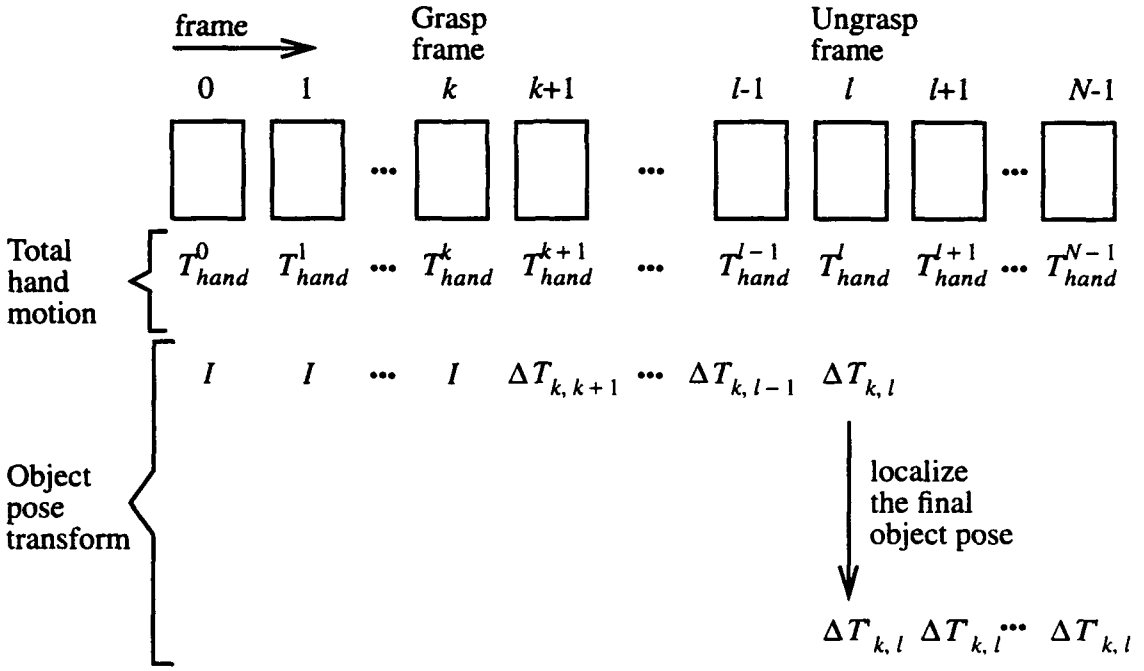


Fig. 16 **Determining the pose of the object throughout the task sequence of N frames**

Based on (5), we can then calculate the object pose transformations at each frame within the manipulation phase as shown in Fig. 16. The pose of the object at the end of the manipulation phase is most likely not very accurate, due to measurement inaccuracies. This pose is refined using a least-squares distance error minimization technique [42].

4.3 Results of applying the 3-pass algorithm

We have applied the 3-pass algorithm on two real 1-tasks to determine the motion breakpoints, identify the grasp employed, and recover the object motion. The first 1-task involves picking up a cylinder from one location and placing it on a different location. The results of the first pass are shown in Fig. 17 and Fig. 18. The pose of the object prior to the performance of the 1-task has been estimated from the range image. As shown in Fig. 18, the motion breakpoints (grasp and ungrasp points) as well as the pregrasp, manipulation, and depart phases are all located.

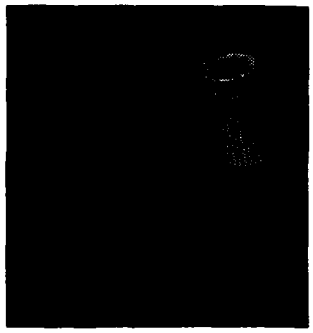


Fig. 17 **Initial pose of the cylinder (1-task #1)**

It is interesting to note the profiles of the average joint flexion angles for each finger and all the fingers (Fig. 19(a)) for this 1-task. As expected, there was very little change in the average flexion angles during the manipulation phase, during which the cylinder was grasped with a power cylindrical grasp. It is also interesting to note that profile of the reciprocal of the average joint angles for all the fingers (Fig. 19(b)) is very similar to that of the fingertip polygon area. This suggests that this may be another metric that can be used in the task segmentation scheme.

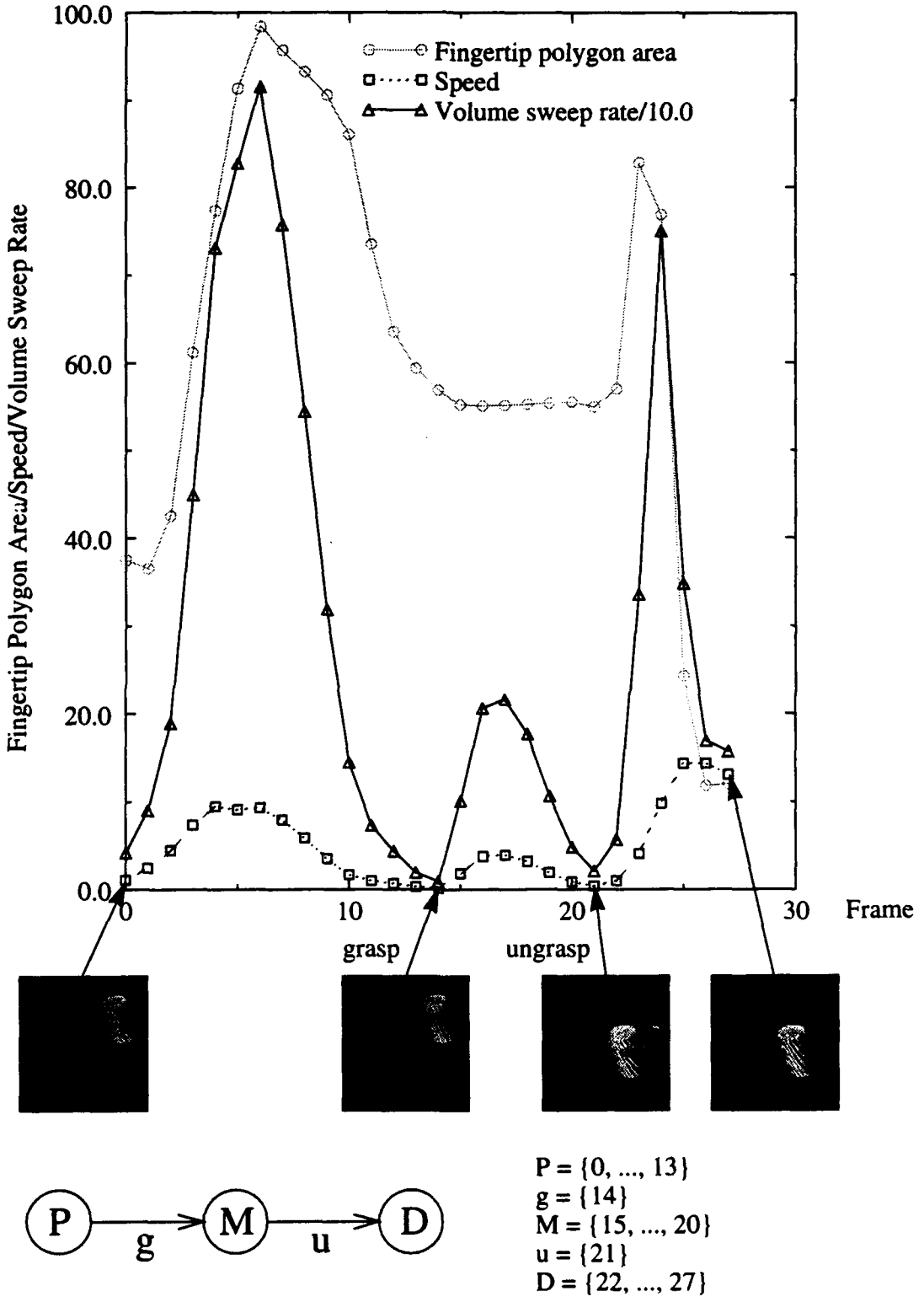


Fig. 18 **Motion profiles and the identified motion breakpoints (1-task #1)**

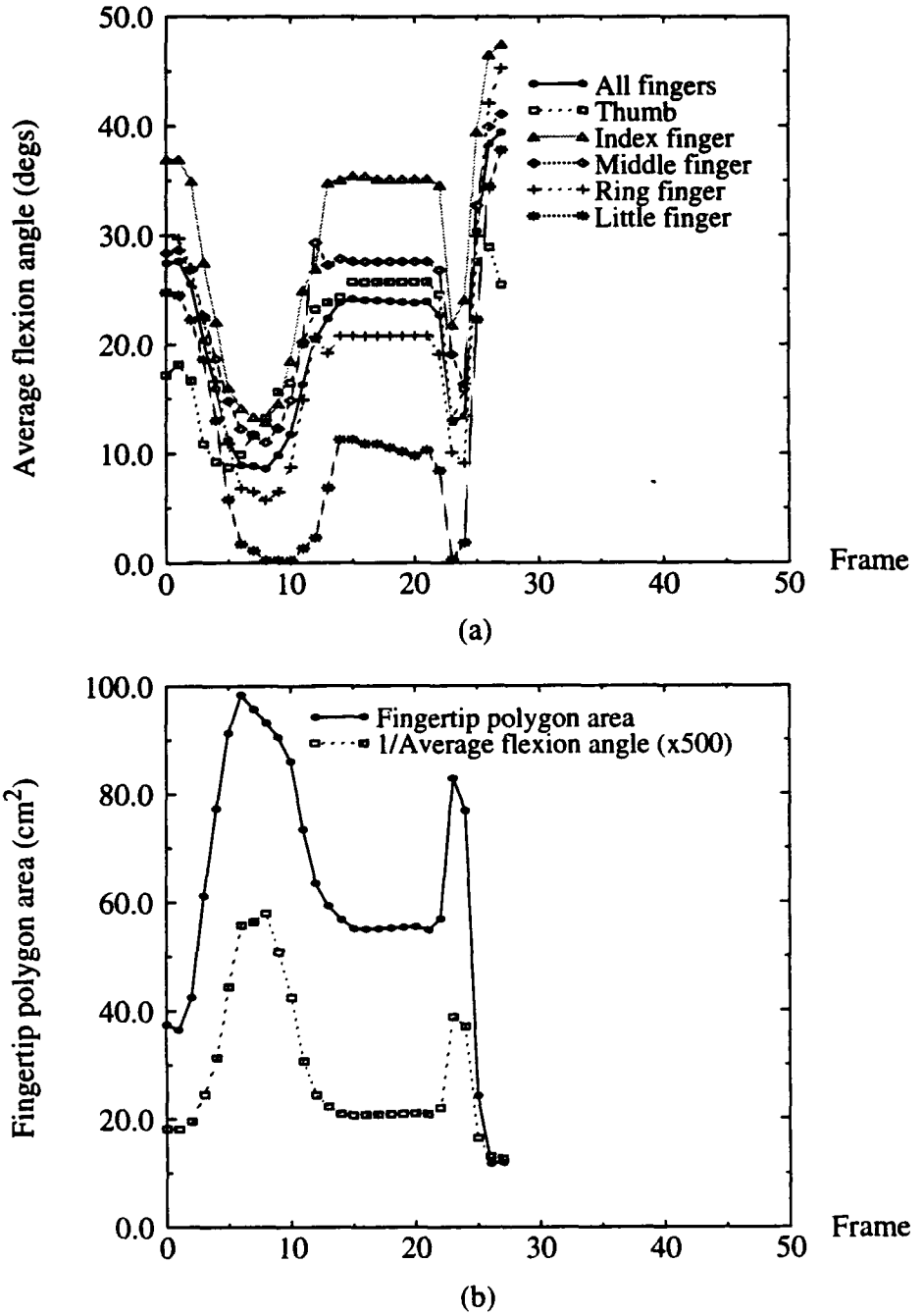
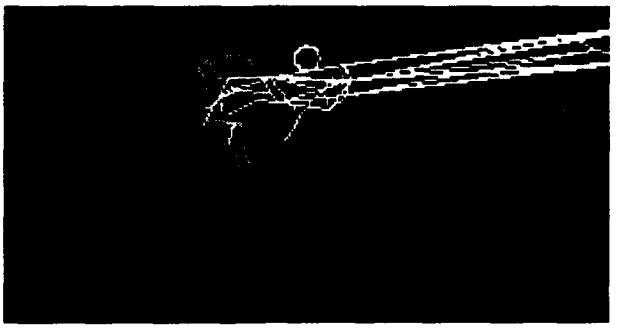
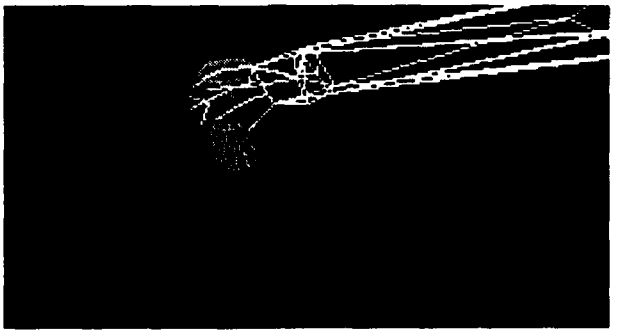


Fig. 19 **Average flexion angle profiles (1-task #1): (a) each and all fingers; (b) comparing the scaled inverse average angle to the fingertip polygon area.**



(a)



(b)

Fig. 20 **Reorienting the grasp in Pass 2: (a) initial pose of the hand relative to the object; (b) final pose of hand relative to cylinder**

Once the hand was reoriented (Fig. 20), the grasp was then correctly identified as a type 2 'coal-hammer' cylindrical grasp[1] using the grasp classification scheme described in [18]. By propagating the extracted object motion during the manipulation phase, the object pose was then estimated (Fig. 21(a)). The pose is subsequently refined (Fig. 21(b)).

1. A 'coal-hammer' cylindrical grasp is one in which the thumb is highly abducted (i.e., significantly deviated from the plane of the palm). This 'coal-hammer' cylindrical grasp is of type 2 because the thumb touches the object. See [18] for more details.

(a) (b)

Fig. 21 **Pose of the cylinder after the task subsequent to Pass 3: (a) pose obtained by successively applying total motion transformations in the manipulation phase; (b) refined pose using the 3DTM program [.?]**

The second 1-task considered is picking up a stick and inserting it through a hole in a castle-shaped object. The two objects involved in this 1-task and the superimposed model of the stick are shown in Fig. 22. Fig. 23 depicts the extracted motion breakpoints and phases of this task.

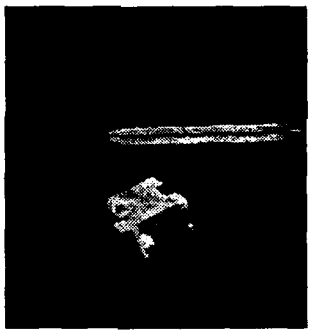


Fig. 22 **Initial pose of the stick (1-task #2)**

As in 1-task #1, the shape of the reciprocal of the average angle profile closely resembles that of the fingertip polygon area (Fig. 24(b)). However, because the stick was held in a precision grasp and the object motion was a combination of translational and rotational motions, there were changes in the average finger joint angles during the manipulation phase (as evidenced in Fig. 24(a)).

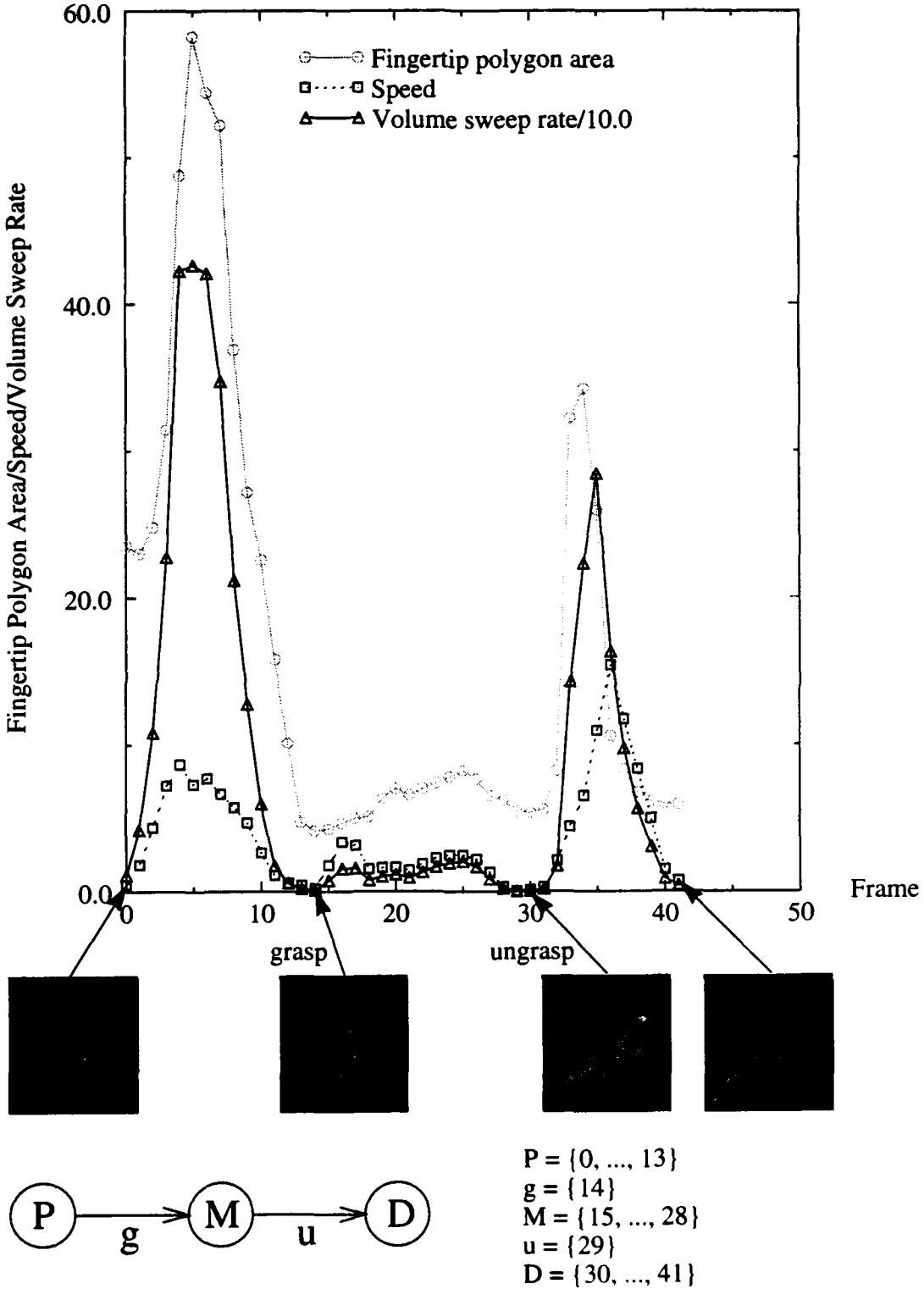


Fig. 23 **Motion profiles and the identified motion breakpoints (1-task #2)**

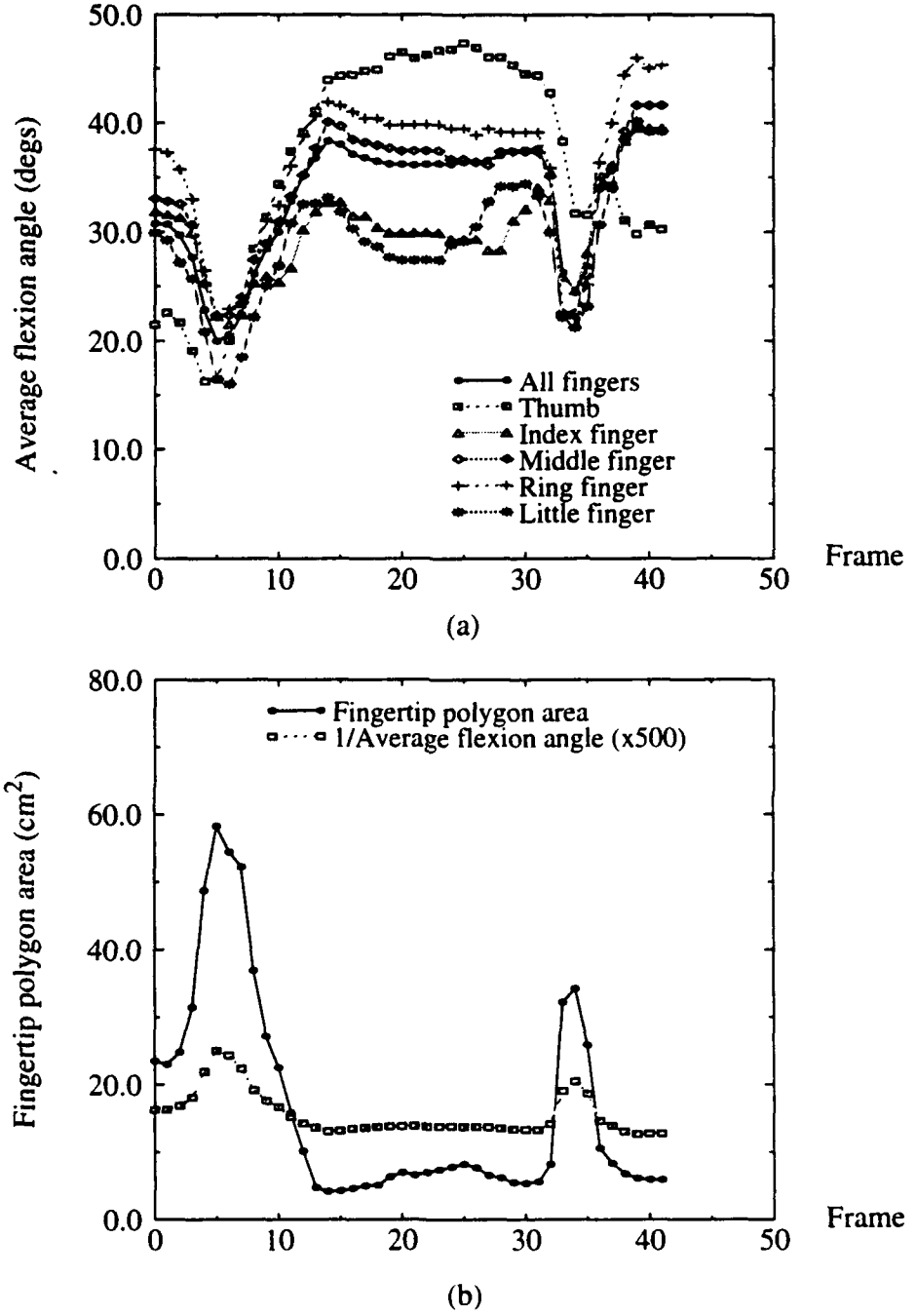
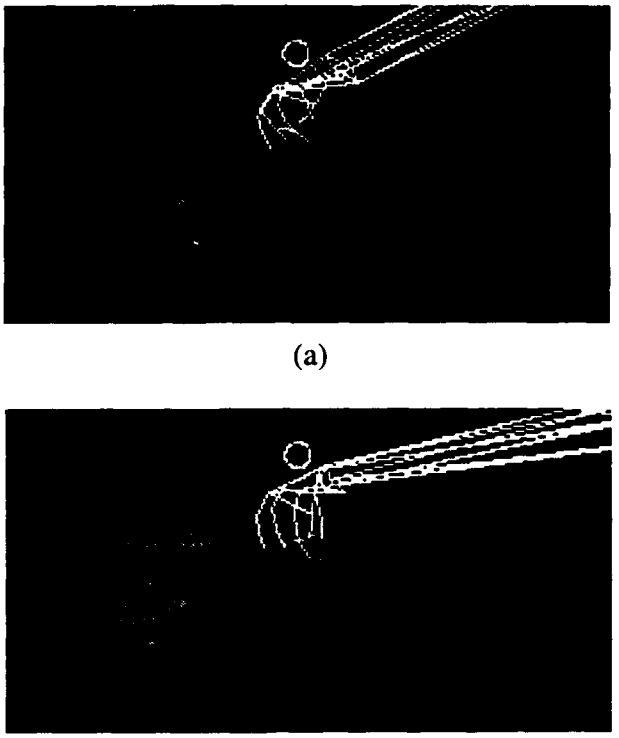


Fig. 24 **Average flexion angle profiles (1-task #2): (a) each and all fingers; (b) comparing the scaled inverse average angle to the fingertip polygon area.**



(a)

(b)

Fig. 25 **Reorienting the grasp in Pass 2: (a) initial pose of the hand relative to the object; (b) final pose of hand relative to stick**

Fig. 25 shows the pose of the hand relative to the stick at the grasp frame before and after reorientation. The grasp was identified as a precision grasp. However, because the middle segments of the four fingers are within the tolerance range of the object (which is set at 1.0 cm), the grasp is classified as a composite nonvolar grasp [18], specifically a prismatic pinch grasp. The grasp that was actually employed in the task is a five-fingered prismatic precision grasp; it can be seen from this result that the while the general grasp classification is correct, the specific category is sensitive to orientation and position errors.

(a) (b)

Fig. 26 **Pose of the stick after the task subsequent to Pass 3: (a) pose obtained by successively applying total motion transformations in the manipulation phase; (b) refined pose using the 3DTM program [42].**

Fig. 26(a) shows the estimated object pose at the end of the task from extracted total motion. Fig. 26(b) shows the refined final object pose.

5 Summary

A task comprises three identifiable phases, namely, the pregrasp, grasp, and manipulation phases. By using the motion profiles of the task, we show that the task can be automatically temporally segmented into these phases. The motion profiles are those of the fingertip polygon area (area of polygon whose vertices are the fingertips) and the speed of the hand motion. We introduce the notion of the *volume sweep rate*, which is the product of the fingertip polygon area and the hand speed. The volume sweep rate profile is also used in the task division algorithm. The successful application of this algorithm on real task examples demonstrates its viability. The temporal task segmentation process is important as it serves as a preprocessing step to the characterization of the task phases. Once the breakpoints have been identified, steps to recognize the grasp and extract the object motion can then be carried out. Two illustrative examples on how these are done were shown.

In addition, it may be useful to characterize the manipulation phase in terms of *total motion* (due to both finger and hand motions), *distal motion* (due to just finger motion) and *proximal motion* (due to just hand motion). While the total motion directly yields the object motion, the proximal and distal motions yield information on which component of the hand/arm motion is contributing to the object motion. This report indicates how these motion transformations can be determined.

Acknowledgments

Many thanks to Mark Wheeler for the use of his 3DTM program and for proofreading this technical report. The idea of plotting out profiles of the average finger joint angles was inspired by discussions with Matt Mason.

Appendix: Determining the transformation between polhemus and rangefinder frames

A.1 Polhemus device mounted at the back of the wrist

In order to merge the hand data from the CyberGlove and Polhemus devices, and range data from the rangefinder, we first recovered the transformation between the CyberGlove/Polhemus and rangefinder reference frames. This is done by laying the CyberGlove/Polhemus devices on the table and taking their range image as well as the Polhemus readings. The required transformation between the two frames are determined from the recovered pose of the Polhemus device in the range image and the Polhemus readings.

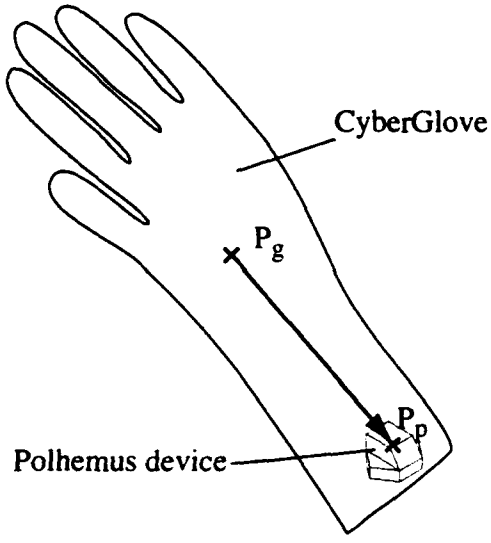


Fig. 27 **The CyberGlove and Polhemus devices and their 3D centroids**

The pose of the Polhemus device is extracted from the range image using the following two steps:

1. *Determine a rough estimate of the pose of the Polhemus device:*

   The 3D centroid of the measurements of the CyberGlove (P_g) is first calculated as shown in Fig. 27. Assuming that the Polhemus device is the brightest region in the image, the intensity image is thresholded and the 3D centroid of the device (P_p) is found. P_p yields a coarse estimate of the device position; by calculating the vector difference between the two centroids (P_p-P_g) and normalizing it, we arrive at a coarse estimate of the device orientation (Fig. 28).

2. *Localization of the Polhemus device using the 3DTM algorithm [42]:*

   The 3DTM (3D template matching) algorithm refines the pose estimate through minimization in a manner similar to deformable templates, active contours, and snakes [20][38]. In this case, the template is derived from the geometric model of the Polhemus device which is created using the geometric modeler Vantage [4]. The formulation of reducing the Euclidean distance between the image 3D points and surface model points is based on the Lorentzian probability distribution; this distribution has the effect of lowering the sensitivity of localization error to object occlusion and extraneous range data. The final localized pose of the Polhemus model is shown superimposed on the image in Fig. 29.

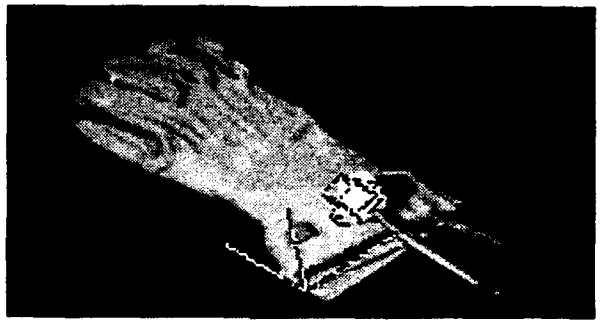


Fig. 28 **Location of the coarsely estimated pose of the Polhemus device**

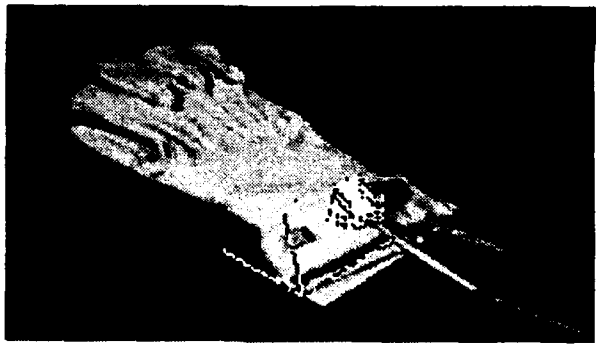


Fig. 29 **Final estimated pose of the Polhemus device**

Let the transformation corresponding to the recovered Polhemus pose in the rangefinder frame be denoted by ${}^{Range}T_{cal}$ and the transformation corresponding to the Polhemus readings be denoted by ${}^{Pol}T_{cal}$. Then the transformation that expresses the coordinates in the Polhemus frame in terms of those in the rangefinder frame is given by

$$ {}^{Range}_{Pol}T = {}^{Range}T_{cal}\ {}^{Pol}T_{cal}^{-1} $$

4.3 Polhemus device mounted at the back of the hand

When the Polhemus device is moved to the back of the hand to reduce the amount of error propagation in the location of the fingers, estimating the initial pose of the Polhemus device

is less straight-forward. The previous method of using the 3D centroids of the CyberGlove and Polhemus devices would not be reliable in this case (Fig. 27) due to their proximity to each other.

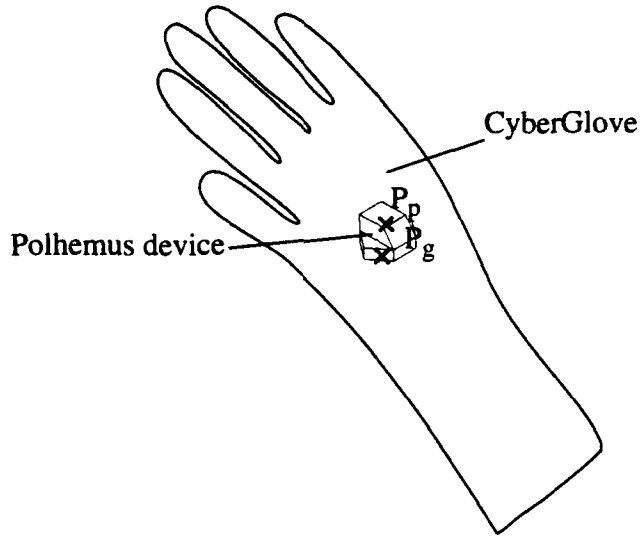


Fig. 30 **The CyberGlove and Polhemus devices and their 3D centroids**

The approach that we took comprises the following steps:

1. *Determine the approximate position of the Polhemus device*

   This is found by determining the 3D centroid of the brightest region in the image (which is assumed to correspond to the Polhemus device).

2. *Determine from principal component analysis the 3D major axis of the region occupied by both the CyberGlove and Polhemus devices*

   This yields the estimated orientation of the Polhemus device. Note that this orientation could be anti-parallel to the correct initial orientation.

3. *Use the 3DTM algorithm to refine the pose of the Polhemus device in the range image*

   However, because of the ambiguity of the initial orientation, we use a two-pass approach: In the first pass, we input the original pose estimation into the program which outputs the refined pose and the average error. Subsequently we modify the refined pose by modifying the orientation by a rotation difference of 180^{o} and use this as input to the second pass. We use the refined pose which corresponds to the lower average error.

Fig. 31 **Initial pose of the Polhemus device**

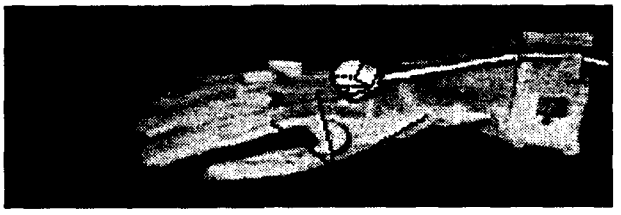


Fig. 32 **Pose immediately after the first pass (switch in orientation)**

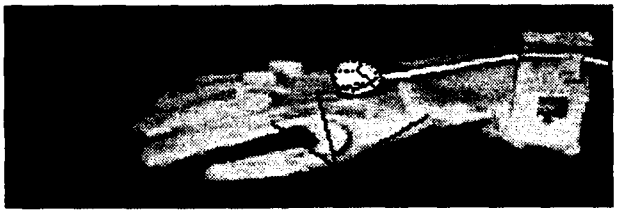


Fig. 33 **Final pose of Polhemus device**

Once the pose of the Polhemus device had been determined, the transformation between the range-finder and Polhemus frames are calculated as in the previous section.

A.3 Linear interpolation of polhemus-to-rangefinder transform

The Polhemus device was originally mounted at the dorsal aspect of the wrist portion of the CyberGlove. This has created serious inaccuracies in the hand and finger locations as the error accumulates from the wrist outwards. These errors include inaccurate hand dimensions and configurations, and errors in joint angle measurement. The Polhemus device uses an ac low-frequency, magnetic field technology to determine the position and orientation of a sensor in relation to a source; nearby ferromagnetic material causes field distortion which would then yield erroneous pose readings. Unfortunately, the equipment that we use has metallic chassis and support frames.

To reduce the inaccuracies due to the compounding effect of angular and configuration errors, we remounted the Polhemus device at the dorsal aspect of the hand. In addition, to compensate for the distorted magnetic field in the workspace (which causes the inaccuracies in the Polhemus readings), we calibrate the Polhemus device at several reasonably well-spaced places (eight) and interpolate between these calibration poses according to spatial proximity. The calibration poses are, however, restricted to those within both the camera and rangefinder views.

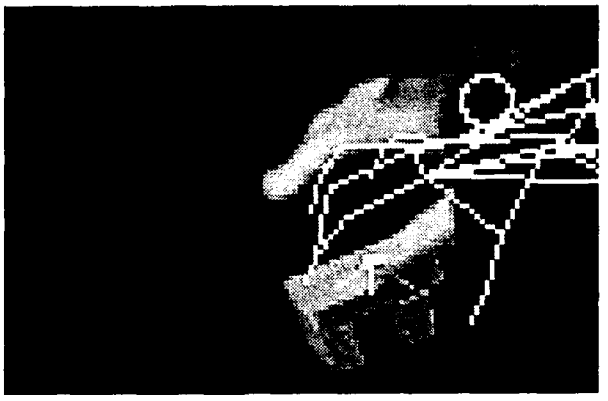


Fig. 34 **Superimposed hand on image in a task sequence with one calibration point**

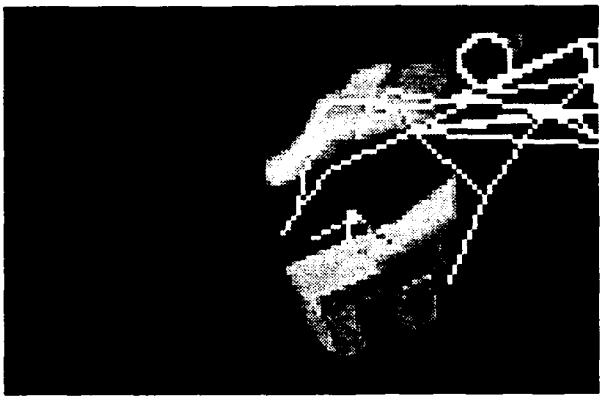


Fig. 35 **Superimposed hand on image in a task sequence with eight calibration points**

Fig. 34 and Fig. 35 show the effect of using several calibration poses as compared to just one.

References

[1] *3Space Isotrak User's Manual*, Polhemus, Inc. Jan. 1992.

[2] M.A. Arbib, T. Iberall, and D.M. Lyons, "Coordinated control programs for movements of the hand," *Hand Function and the Neocortex*, eds. A.W. Goodwin, and I. Darian-Smith, Springer-Verlag, 1985, pp. 111-129.

[3] H. Asada, and Y. Asari, "The direct teaching of tool manipulation skills via the impedance identification of human motions," *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1988, pp. 1269-1274.

[4] P. Balakumar, J.C. Robert, R. Hoffman, K. Ikeuchi, and T. Kanade, *VANTAGE: A Frame-based Geometric Modeling System - Programmer/User's Manual*, Carnegie Mellon University, Dec. 1988.

[5] C. Bard, J. Troccaz, and G. Vercelli, "Shape analysis and hand preshaping for grasping," *Proc. IEEE/RSJ Int'l Workshop on Intelligent Robots and Systems*, 1991, pp. 64-69.

[6] H.J. Buchner, M.J. Hines, and H. Hemami, "A dynamic model for finger interphalangeal coordination," *Journal of Biomechanics*, vol. 21, no. 6, 1988, pp. 459-468.

[7] Cyber*Glove*^{TM} *System Documentation*, Virtual Technologies, June 1992.

[8] R. Finkel, R. Taylor, R. Bolles, R. Paul, and J. Feldman, *AL: A programming system for automation*, Tech. Rep. AIM-177, Stanford University, Artificial Intelligence Lab., 1974.

[9] W.A. Gruver, B.I. Soroka, J.J. Craig, and T.L. Turner, "Evaluation of commercially available robot programming languages," *Proc. 13th Int'l Symp. on Industrial Robots*, 1983, pp. 12-58.

[10] T. Hamada, K. Kamejima, and I. Takeuchi, "Image based operation; A human-robot interaction architecture for intelligent manufacturing," *Proc. 15th Conf. of IEEE Industrial Electronics Society*, 1989, pp. 556-561.

[11] H. Hashimoto, and M. Buss, "Skill acquisition for the Intelligent Assisting System using Virtual Reality Simulator," *Proc. 2nd Int'l Conf. on Artificial Reality and Tele-Existence (ICAT '92)*, Tokyo, Japan, 1992, pp. 37-46.

[12] S. Hirai, and T. Sato, "Motion understanding for world model management of telerobot," *Proc. 5th Int'l Symp. on Robotics Research*, 1989, pp. 5-12.

[13] W. Iba, *Acquisition and improvement of human motor skills: learning through observation and practice*, Tech. Rep. RIA-91-29, NASA Ames Research Center, Artificial Intelligence Research Branch, Oct. 1991.

[14] T. Iberall, J. Jackson, L. Labbe, and R. Zampano, "Knowledge-based prehension: capturing human dexterity," *Proc. IEEE Int'l Conf. of Robotics and Automation*, 1988, pp. 82-87.

[15] K. Ikeuchi, and T. Suehiro, "Towards an Assembly Plan from Observation, Part I: Assembly task recognition using face-contact relations (polyhedral objects)," *Proc. Int'l Conf. on Robotics and Automation*, 1992, pp. 2171-2177. A longer version is available as Tech. Rep. CMU-CS-91-167, Carnegie Mellon University, Aug. 1991.

[16] M. Jeannerod, "Intersegmental coordination during reaching at natural visual objects," *Attention and Performance IX*, Long, J., and Baddley, A. (eds.), Erlbaum, Hillsdale, NJ. 1981, pp. 153-168.

[17] M. Jeannerod, "The timing of natural prehension movements," *Journal of Motor Behavior*, vol. 16, no. 3, 1984, pp. 235-254.

[18] S.B. Kang, and K. Ikeuchi, "Grasp recognition using the contact web", *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 1992, pp. 194-201. A longer version is available as Tech. Rep. CMU-RI-TR-91-24, Carnegie Mellon University, Nov. 1991.

[19] S.B. Kang, and K. Ikeuchi, "A grasp abstraction hierarchy for recognition of grasping tasks from observation," to appear in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, July, 1993.

[20] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int'l Journal of Computer Vision*, vol. 2, no. 1, 1987, pp. 322-331.

[21] *Knowledge Craft Manual - Vol. I: CRL Technical Manual*, Carnegie Group, Inc., 1989.

[22] T. Kuniyoshi, M. Inaba, and H. Inoue, "Teaching by showing: Generating robot programs by visual observation of human performance," *Proc. 20th Int'l Symp. on Industrial Robots*, 1989, pp. 119-126.

[23] P. Lammineur, and O. Cornillie, *Industrial Robots*, Pergammon Press, 1984, pp. 43-54.

[24] J.M.F. Landsmeer, "Power grip and precision handling," *Ann. Rheum. Dis.*, Vol. 21, 1962, pp. 164-170.

[25] H. Liu, T. Iberall, and G.A. Bekey, "The multi-dimensional quality of task requirements for dextrous robot hand control," *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1989, pp. 452-457.

[26] T. Lozano-Perez, "Automatic planning of manipulator transfer movements," *IEEE Trans. on Systems, Man and Cybernetics*, SMC-11(10), 1981, pp. 681-689.

[27] D.M. Lyons, "A simple set of grasps for a dextrous hand," *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1985, pp. 588-593.

[28] C.L. MacKenzie, and J. Van den Biggelaar, "The effects of visual information, object motion and size on reaching and grasping kinematics," *Soc. for Neuroscience Abstracts*, vol. 13, part 1, 1987, pg. 351.

[29] R.G. Marteniuk, and S. Athenes, "Characteristics of natural prehension movements for objects of verying size," *Soc. for Neuroscience Abstracts*, vol. 12, part 2, 1986, pg. 970.

[30] R.G. Marteniuk, C.L. MacKenzie, M. Jeannerod, S. Athenes, and C. Dugas, "Constraints on human arm movement trajectories," *Canadian Journal of Psychology*, vol. 41, no. 3, 1987, pp. 365-378.

[31] M.T. Mason, and J.K. Salisbury, *Robot Hands and the Mechanics of Manipulation*, MIT Press, 1985.

[32] P. Morasso, "Spatial control of arm movements," *Experimental Brain Research*, vol. 42, no. 1, 1981, pp. 223-227.

[33] K. Perlin, J.W. Demmel, and P.K. Wright, "Simulation software for the Utah/MIT dextrous hand," *Robotics and Computer-Integrated Manufacturing*, vol. 5, no. 4, 1989, pp. 281-292.

[34] P.K. Pook, and D.H. Ballard, "Recognizing teleoperated manipulations," *Proc. IEEE Int'l Conf. on Robotics and Automation*, vol. 2, 1993, pp. 578-585.

[35] H. Rijpkema, and M. Girard, "Computer animation of knowledge-based human grasping," *Computer Graphics*, vol. 25, no. 4, 1991, pp. 339-348.

[36] J.M. Rubin, and W.A. Richards, *Boundaries of visual motion*, Tech. Rep. AIM-835, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Apr. 1985.

[37] T. Takahashi, and H. Ogata, "Robotic assembly operation based on task-level teaching in virtual reality," *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1992, pp. 1083-1088.

[38] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: Recovering 3D shape and nonrigid motion," *Artificial Intelligence*, vol. 36, 1988, pp. 91-123.

[39] R. Tomovic, G.A. Bekey, and W.J. Karplus, "A strategy for grasp synthesis with multifingered robot hands," *Proc. IEEE Int'l Conf. of Robotics and Automation*, 1987, pp. 83-89.

[40] A.M. Wing, and C. Fraser, "The contribution of the thumb to reaching movements," *Quarterly Journal of Experimental Psychology*, vol. 35A, 1983, pp. 297-309.

[41] A.M. Wing, A. Turton, and C. Fraser, "Grasp size and accuracy of approach in reaching," *Journal of Motor Behavior*, vol. 18, no. 3, 1986, pp. 245-260.

[42] M.D. Wheeler, and K. Ikeuchi, *Towards a Vision Algorithm Compiler for recognition of partially occluded 3-D objects*, Tech. Rep. CMU-CS-92-185, Carnegie Mellon University, Nov. 1992.